IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of:

Shai Mohaban, et al.

Serial No.: 09/347,438

Filed on: July 2, 1999

For: METHOD AND APPARATUS FOR CREATING POLICIES FOR POLICY-BASED MANAGEMENT OF QUALITY OF SERVICE TREATMENTS OF NETWORK DATA TRAFFIC FLOWS

Confirmation No.: 3850

Group Art Unit No.: 2157

Examiner: Barbara N. Burgess

Customer No.: 29989

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CORRECTED APPEAL BRIEF

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed on March 23, 2006 and in response to the Office communication mailed August 24, 2006, for which the one (1) month shortened statutory period for reply ends on September 25, 2006.

I. REAL PARTY IN INTEREST

The assignee Cisco Technology, Inc., and its parent corporation Cisco Systems, Inc., San Jose, California, are the real parties in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-4, 6-17, and 19-30 are pending in this application, were finally rejected and are the subject of this appeal. Claims 5 and 18 were canceled during prosecution.

IV. STATUS OF AMENDMENTS

No amendments were filed after the final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The application contains independent Claims 1, 19, 20, 21, 29, and 30. These independent claims recite similar limitations, except in the context of an apparatus, a method and a computer-readable medium. There are no claims in means-plus-function format.

The independent claims are directed generally to an approach for automatically setting quality of service (QoS) values in packet data that is traveling through a router, switch or other device in a network, based on application code points, a stored policy and the type of traffic (for example, Specification, p. 8; p. 15, lines 14-16; p. 16 line 1 to p. 18 line 17; Abstract). For example, in claim 1, application information that defines one or more traffic flows associated with one or more message types generated by an application program is received (for example, Specification, p. 16 line 4-10; p. 19 line 16 to p. 21 line 19; FIG. 4, 426; FIG. 7A, 702). The information identifies one or more points at which an application generates the traffic flows (for example, Specification, p. 16 line 4-10; p. 17 line 1-5; p. 19 line 16 to p. 21 line 19; FIG. 4, 426). Device information, which defines one of more quality of service treatments that the particular network device may apply to data processed by the particular network device, is received (for example, p. 14 line 16 to p. 15 line 3). Based on the device information and the application information, one or more processing policies that associate the traffic flows with the quality of service treatments are determined (for example, p. 14 lines 16-23; p. 16, lines 11-20; FIG. 7A, 704; p. 20 line 24 to p. 21 line 14; FIG. 4, 4206, 410).

Mappings of the application points to the quality of service treatments are stored (for example, Specification p. 14 line 7-9; FIG. 6A, 604; p. 16 lines 21-26; FIG. 7A, 706). The mappings may be used with the processing policies to generate the quality of service value when the application program generates traffic flows of one of the message types (for example, Specification p. 15 lines 9-23; FIG. 6B, 609; p. 17 lines 1-23; FIG. 7B, 708 to 714). The QoS value is generated based on the mappings, before transmitting the traffic flows of one of the

message types to the network ((for example, Specification p. 17, lines 14-22; FIG. 7B, 710 to 714). Further, the processing policies are enforced at the network device in response to receiving traffic from the application program that matches the traffic flow type (for example, Specification p. 15 lines 9-23; FIG. 6B, 609). For example, enforcing one of the processing policies comprises requesting, using an application QoS policy element that is coupled to the application program, an operating system function to modify a packet of the traffic flows using a policy element that requests a different operating system function according to the operating system then in use (for example, Specification p. 15, lines 14-23; FIG. 6B, 609; p. 18, lines 1-8). At the network device, in response to receiving traffic from the application program that matches the traffic flow type and in response to the operating system function, a portion of the packet is modified to activate a quality of service treatment of the network device (for example, Specification at p. 15 lines 14-23; p. 18, lines 8-17; FIG. 7B, 718).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 2, 6, 9, 14-16, 20-22, 24-25, and 27-30 stand rejected under 35 USC §103(a) as allegedly unpatentable over Martin (U.S. Pat. No. 6,154,776) in view of Haddock et al. (U.S. Pat. No. 6,104,700) and further in view of Colley et al. (U.S. Pat. No. 6,650,644 B1).

2. Claims 3, 4, and 23 stand rejected under 35 USC §103(a) as allegedly unpatentable over Martin (U.S. Pat. No. 6,154,776) in view of Haddock et al. (U.S. Pat. No. 6,104,700) and further in view of Colley et al. (U.S. Pat. No. 6,650,644 B1) and in further view of Chapman et al. (U.S. Pat. No. 6,028,842).

3. Claims 10-11, 17, 19, and 26 stand rejected under 35 USC §103(a) as being unpatentable over Martin in view of Haddock et al. and in further view of Colley et al. and in further view of Chapman in further view of Mohaban et al. (U.S. Pat. No. 6,028,842) in further view of Mohaban et al (U.S. Pat. No. 6,463,470).

4. The Office Action has rejected Claims 12 and 13 under 35 USC §103(a) as being unpatentable over Martin in view of Haddock et al. and in further view of Colley et al. and in further view of Schwaller et al. (U.S. Pat. No. 6,061,725).

VII. ARGUMENTS

To establish a *prima facie* case of obviousness under 35 U.S.C. § 103(a), the references cited and relied upon must teach or suggest all the claim features. In addition, a sufficient factual basis to support the obviousness rejection must be proffered. *In re Freed*, 165 USPQ 570 (CCPA 1970); *In re Warner*, 154 USPQ 173 (CCPA 1967); *In re Lunsford*, 148 USPQ 721 (CCPA 1966). In the present case, the references lack particular claim features, and the Office Action does not articulate a proper rationale to combine the references.

A. CLAIMS 1, 2, 6, 9, 14-16, 20-22, 24-25, AND 27-30—MARTIN IN VIEW OF HADDOCK IN VIEW OF COLLEY

Claims 1, 2, 6, 9, 14-16, 20-22, 24-25, and 27-30 stand rejected under 35 USC §103(a) as allegedly unpatentable over Martin (U.S. Pat. No. 6,154,776) in view of Haddock et al. (U.S. Pat. No. 6,104,700) and further in view of Colley et al. (U.S. Pat. No. 6,650,644 B1). The rejections should be reversed.

All elements, steps or limitations of a claim must be taught, disclosed or suggested in the cited references, and the Office Action must provide a supportable rationale to combine the references, to support a *prima facie* case of obviousness under §103(a). Here, numerous specific claim features are missing from the references, when the references are closely studied and considered.

For example, the references fail to provide "receiving application information that defines one or more traffic flows **associated with one or more message types generated by an application program**, including **information identifying one or more points at which an application generates the traffic flows**," as recited in all the rejected independent claims (1, 20, 21, 29, 30). The Office Action contends that Martin shows the quoted feature at 2:25-28, 2:44-47, 3:2-4, 3:9-15, 3:34-39, 3:48-49, 3:55-59, 4:1-5, 4:13-15, 4:33-38, 4:52-55, 5:1-3, 9:65-67, and 10:1-2. This is incorrect.

Martin 2:25-28 states that a QoS allocation to an information flow belonging to an "entity" is possible. Martin 2:44-47 states that an "entity" can include an application (and, at

4:37-40, a user, equipment, or group), **but not message types generated by an application program**, as claimed. Martin 3:2-4 states that an entity "will seek a presence on the network to establish an information flow," which is irrelevant to the feature at issue.

Martin 3:9-15 refers to the problem of *a priori* establishment of QoS when an application has a large number of instances, **but has no reference to message types generated by an application program.** "Instances," in this context, must be interpreted using the plain and ordinary technical meaning of the term to refer to executable instances, and not message types, as claimed.

Martin 3:34-39 describes a method of "detecting a new instance of an entity associated with a network flow," but has no teaching or suggestion of **message types generated by an application program**, or **information identifying one or more points at which an application generates the traffic flows**, as claimed. For example, with Martin, the same application instance sending multiple different message types would be detected once and associated with one QoS policy. With Applicants' claims, **each message type potentially maps to a different QoS policy**. Martin has no way to apply QoS at the message level.

Similarly, Martin 3:48-49 refers to "detection of a new instances of an entity associated with a flow," but has no teaching or suggestion of **message types generated by an application program**, or **information identifying one or more points at which an application generates the traffic flows**, as claimed.

Martin 3:55-59 states that different ways can be used to detect a new instance of an entity, but does not teach message types or application points. Martin 4:1-5 refers to using a default rule for a new instance of an entity. Martin 4:13-15 states that flow parameters can be used as a key. Martin 4:33-38 defines an "entity" as stated above, and the definition is specific and does not include or anticipate message types of an application or points in an application program at which traffic flows are generated.

Martin 4:52-55 states that a controller responsive to a new instance of an entity can be used, but does not teach or suggest the quoted feature above. Martin 5:1-3 states that a QoS

mechanism can perform the detection, but does not teach or suggest the quoted feature above. Martin 9:65-67 to 10:1-2 describes use of a user profile or application profile, but does not teach or suggest the quoted feature above.

Further, the references fail to provide "creating and storing one or more mappings of the application points to the quality of service treatments that may be used with the processing policies to generate the quality of service value when the application program generates **traffic flows of one of the message types**," as recited in all the independent claims. The Office Action contends that Martin shows the quoted feature at 3:55-56, 4:20-25, 4:64-67, 5:5-7, 8:38-40, 8:47-50, 10:3-5, 10:34-35, 10:40-46, and 13:50-53. This is incorrect.

Martin 3:55-56 states that different ways can be used to detect a new instance of an entity, but does not teach message types or application points.

Martin 4:20-25 states that a QoS identifier can represent a flow-entity binding and can be used to retrieve a QoS definition, but has no teaching or disclosure of **message types** or mappings that can be used in response to particular message types.

Martin 4:64-67 states that a directory service maintains a mapping between a flow and an entity and for QoS identifications and definitions, but has no teaching or disclosure of **message types** or mappings that can be used in response to particular message types of applications. Martin is concerned only with applications as a whole, not different message types that the applications can generate.

Martin 5:5-7 describes caching stored mappings, but has no teaching or disclosure of **message types** or mappings that can be used in response to particular message types of applications.

Martin 8:38-40 states that an entity entry can define a mapping of an entity to one or more flow parameters. The preceding sentence states that such parameters include an allocated IP address or port. Nothing in Martin 8:38-40 refers to establishing mappings between message types of an entity and QoS values, as claimed.

Martin 8:47-50 describes a data structure that maintains mappings of flows and entities, QoS identifiers, and definitions, but has no teaching or disclosure of **message types** or mappings that can be used in response to particular message types of applications.

Martin 10:3-5 states that an application profile can store parameters for a flow associated with an application, group of users, or group of services, but has no teaching or disclosure of **message types** or mappings that can be used in response to particular message types of applications. Nothing in Martin 10:3-5 can be interpreted to provide message-level flow mappings.

Martin 10:34-35 refers to returning a QoS identification as a value or list. Martin 10:40-46 refers to a QoS definition that can have mappings or links to rules. Martin 13:50-53 refers to retrieving QoS definitions from a directory. But none of the cited passages in Martin have any teaching or disclosure of **message types** or mappings that can be used in response to particular message types of applications. Martin is concerned only with applications as atomic entities, not applying different QoS to different types of messages.

For all the foregoing reasons, Martin fails to teach or disclose several features of all the independent claims. The contentions of the Office Action are unsupported in Martin. Therefore, any combination of Martin with Haddock and Colley cannot provide the complete invention as claimed. Accordingly, the Office Action fails to present a *prima facie* case of unpatentability under 35 U.S.C. §103(a) with respect to the independent claims.

Each of the other pending claims among claims 1, 2, 6, 9, 14-16, 20-22, 24-25, and 27-30 includes the foregoing features directly, or depends from an independent claim that does contain the foregoing features. Therefore, the Office Action fails to present a *prima facie* case of unpatentability under 35 U.S.C. §103(a) with respect to each of claims 1, 2, 6, 9, 14-16, 20-22, 24-25, and 27-30.

B. CLAIMS 3, 4, AND 23—MARTIN IN VIEW OF HADDOCK, COLLEY, AND CHAPMAN

Claims 3, 4, and 23 stand rejected under 35 USC §103(a) as allegedly unpatentable over Martin (U.S. Pat. No. 6,154,776) in view of Haddock et al. (U.S. Pat. No. 6,104,700) and further in view of Colley et al. (U.S. Pat. No. 6,650,644 B1) and in further view of Chapman et al. (U.S. Pat. No. 6,028,842). The rejections should be reversed.

Claim 3 recites the method of claim 1, further comprising "creating and storing one or more classes that classify the traffic flows, **each of the classes associated with one or more of the message types;** based on the device information and the classes of the traffic flows, determining one or more processing policies that associate the traffic flows with the quality of service treatments." Claim 23 corresponds in scope to claim 3.

As described extensively above, Martin fails to provide any teaching of associating QoS policies or values with **message types**, as opposed to application entities. Claim 3, and independent claim 1, both expressly refer to message types. Therefore, any combination of Martin with the other references cannot provide the complete claimed subject matter.

Haddock, Colley, and Chapman fail to cure the deficiency of Martin. Indeed, the Office Action provides no citation to any of the references to provide message types. Instead, the Office Action relies on Chapman alone to show "the use and advantages for classifying traffic flows." However, the use and advantages for classifying traffic flows is not what is claimed. Rather, the claim recites creating and storing classes associated with message types in particular. Generalized knowledge among those skilled in the art about classifying traffic flows would not necessarily cause someone of skill in the art in 1999 or earlier to classify traffic flows **based upon message types of applications**.

Chapman 1:33-34 referes to bandwidth control to ensure that traffic classes are not starved. Chapman 2:1-3, 2:6-7, 2:27-28, 2:40-43, and 2:50-53 refer to traffic classification in a vague, non-specific way, and certainly have no suggestion to perform traffic classification based upon application message types, as claimed.

Claim 4 recites the method of claim 1, wherein receiving application information comprises receiving one or more application code points that represent traffic flow types. The Office Action relies on Chapman 3:46-48, 3:51-55, 3:63, 3:65-66, 4:3-5, 4:8-10, 4:12-14, 4:19-22, and 4:29-31. However, these passages merely describe associating different traffic classes with different groups of applications. The term "application code point," as recited in the claim and as used in Applicants' specification, is not found in any of the cited passages. For example, Applicants' specification uses "application code point" to refer to a point in an application that generates a particular message type. Chapman has no equivalent disclosure.

Further the rejection under 35 USC §103(a) is deemed moot in view of Applicant's comments above regarding claim 1 and the independent claims. Claims 3-4, and 23 are dependent upon independent claims 1 and 21, respectively.

When a §103(a) rejection is based upon multiple references, the Office Action must provide proper evidence of a suggestion or motivation among those of skill in the art, at the time the invention was made, to combine the references to result in the claimed subject matter. Here, the Office Action contends that "the use and advantages for using application code points to represent traffic flow types" was well known, and therefore a skilled artisan "would have found it obvious to implement or incorporate application code points in Martin's method to allocate bandwidth and implement an admission control policy for classes before delivering a packet." The rationale of the Office Action fails for three reasons. First, as stated above, Chapman has no specific description of "application code points." Second, merely knowing the use and advantages of traffic classification in a general way cannot motivate the specific claimed combination that uses application code points; such general knowledge does not mean that a skilled artisan would think of **every possible method of traffic classification**, including Applicants' method. Third, the claimed invention does not recite allocating bandwidth and implementing an admission control policy for classes before delivering a packet.

C. CLAIMS 10-11, 17, 19, AND 26—MARTIN, HADDOCK, COLLEY, CHAPMAN, MOHABAN

Claims 10-11, 17, 19, and 26 stand rejected under 35 USC §103(a) as being unpatentable over Martin in view of Haddock et al. and in further view of Colley et al. and in further view of Chapman in further view of Mohaban et al. (U.S. Pat. No. 6,028,842) in further view of Mohaban et al (U.S. Pat. No. 6,463,470). The rejections should be reversed.

Under the "common ownership exception" codified at 35 USC §103(c)(1), Mohaban '470 does not qualify as a prior art reference for this application. Mohaban '470 qualifies as a reference, if at all, only under 35 USC §102(e), not §102(b). Applicants have previously stated in a reply, and now re-state, that both Mohaban '470 and the present application were owned by, subject to an obligation of assignment to, or assigned to the same person—Cisco Technology, Inc.—at the time of the invention of the subject matter of the present application. Because both Mohaban '470 and the present application were commonly owned, Mohaban '470 is disqualified as prior art and must be withdrawn as a reference.

In a previous Office Action, the Examiner argued that prior section 35 USC §103(c) is inapplicable to the present application, because the filing date of the present application (July 2, 1999) predates the enactment on November 29, 1999 of prior section 35 USC §103(c). However, that argument is untenable in view of the amendment to 35 U.S.C. 103(c) by the Cooperative Research and Technology Enhancement Act of 2004 (CREATE Act). As stated in the USPTO's own online "frequently asked questions" page (http://www.uspto.gov/web/offices/dcom/olia/aipa/103cfaq.htm), the "common ownership exception" applies to this case:

> The CREATE Act was enacted on December 10, 2004, and is effective for all patents, including reissued patents, granted on or after the enactment date. In other words, it is effective for all patent applications pending on or after December 10, 2004. The CREATE Act also effectively makes the 1999 amendment to 35 U.S.C. 103(c) applicable to any applications filed prior to November 29, 1999 and were pending on December 10, 2004. The 1999

amendment to 35 U.S.C. 103(c) added certain commonly owned or assigned prior art under 35 U.S.C. 102(e) to the prior art under 35 U.S.C. 102(f) and (g) that can be disqualified under 35 U.S.C. 103(c).

Because Mohaban '470 is unavailable as a reference, the Office Action fails to present a *prima facie* case of obviousness under 35 USC §103(a).

Even if Mohaban '470 is available, the rejection under 35 USC §103(a) is deemed moot in view of Applicant's comments above regarding the independent claims. Each of the pending claims 10-11, 17, 19, and 26 includes the features of claim 1 discussed above directly, or depends from an independent claim that does contain the features of claim 1 discussed above. Thus, because the base reference (Martin) lacks features of the base independent claims, a combination of Martin, Haddock, Colley, Chapman, and Mohaban fails to provide the complete claimed subject matter of claims 10-11, 17, 19, and 26. Therefore, the Office Action fails to present a *prima facie* case of unpatentability under 35 U.S.C. §103(a) with respect to each of claims 1, 2, 6, 9, 14-16, 20-22, 24-25, and 27-30.

D. CLAIMS 12-13—MARTIN, HADDOCK, COLLEY, AND SCHWALLER

The Office Action has rejected Claims 12 and 13 under 35 USC §103(a) as being unpatentable over Martin in view of Haddock et al. and in further view of Colley et al. and in further view of Schwaller et al. (U.S. Pat. No. 6,061,725).

The rejection under 35 USC §103(a) is deemed moot in view of Applicant's comments regarding Claims 1, 20, 21, 29, and 30, above. Claims 12-13 are dependent upon Claim 1. Because the base references do not provide all features of Claim 1, a combination with Schwaller et al. cannot provide the complete subject matter recited in Claims 12-13. Therefore, Applicant respectfully requests reversal of the rejection under 35 USC §103(a).

/ / /

/ / /

/ / /

VII. CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, Applicants respectfully submit that the rejections of the claims are unsupported in law or in fact. Appellants therefore respectfully request that the Honorable Board reverse the rejections of the claims.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: September 25, 2006

/ChristopherJPalermo#42056/
Christopher J. Palermo
Reg. No. 42,056

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Telephone No.: (408) 414-1080 ext. 214
Facsimile No.: (408) 414-1076

CLAIMS APPENDIX

1. A method of selectively establishing a quality of service value for a particular network device in a network that comprises a plurality of other heterogeneous network devices, comprising the steps of:

receiving application information that defines one or more traffic flows associated with one or more message types generated by an application program, including information identifying one or more points at which an application generates the traffic flows;

receiving device information that defines one of more quality of service treatments that the particular network device may apply to data processed by the particular network device;

based on the device information and the application information, determining one or more processing policies that associate the traffic flows with the quality of service treatments;

creating and storing one or more mappings of the application points to the quality of service treatments that may be used with the processing policies to generate the quality of service value when the application program generates traffic flows of one of the message types;

causing generation of the quality of service value, wherein the generation of the quality of service value is based on said one or more mappings and is performed before transmitting said traffic flows of one of the message types to said network;

enforcing one of the processing policies at the network device in response to receiving traffic from the application program that matches the traffic flow type; and

wherein enforcing one of the processing policies comprises:

requesting, using an application QoS policy element that is coupled to the application program, an operating system function to modify a packet of the traffic flows using a policy element that requests a different operating system function according to the operating system then in use; and

at the network device, in response to receiving traffic from the application program that matches the traffic flow type and in response to the operating

system function, modifying a portion of the packet to activate a quality of service treatment of the network device.

2. A method as recited in Claim 1, further comprising:
storing the mappings in a repository that is accessible by the application program;
storing both the application information and the device information in the repository; and
converting the mappings into one or more settings of the network device.

3. A method as recited in Claim 1, further comprising:
creating and storing one or more classes that classify the traffic flows, each of the classes associated with one or more of the message types;
based on the device information and the classes of the traffic flows, determining one or more processing policies that associate the traffic flows with the quality of service treatments.

4. A method as recited in Claim 1, wherein receiving application information comprises receiving one or more application code points that represent traffic flow types.

5. (Canceled)

6. A method as recited in Claim 1, wherein creating and storing one or more mappings comprises creating and storing one or more policies, concerning network processing of traffic flows generated by the application program, in the repository.

7. A method as recited in Claim 1, wherein creating and storing one or more mappings comprises creating and storing one or more policies, concerning network processing of traffic flows generated by the application program, in a policy store that is coupled to the repository.

8. A method as recited in Claim 1, wherein creating and storing one or more mappings comprises creating and storing one or more policies, concerning network processing of traffic flows generated by the application program, in a directory.

9. A method as recited in Claim 1, wherein creating and storing one or more mappings comprises creating and storing one or more policies, concerning network processing of traffic flows generated by the application program, in a policy server coupled to a Lightweight Directory Access Protocol directory that comprises the repository.

10. A method as recited in Claim 1, wherein creating and storing one or more mappings further comprises creating and storing, in the repository, one or more mappings of Application Code Points of the application program to one or more Differential Services Code Points of a protocol associated with the network device.

11. A method as recited in Claim 1, wherein creating and storing one or more mappings further comprises generating one or more messages in RSVP+ () and communicating the messages to the network device.

12. A method as recited in Claim 1, wherein receiving application information comprises receiving application information that defines one or more traffic flows generated by an application program, including information identifying one or more points at which an application generates the traffic flows, from a first individual having responsibility for managing enterprise applications in the network, and not from one having responsibility for managing the network.

13. A method as recited in Claim 12, wherein receiving device information comprises receiving device information that defines one of more quality of service treatments that the network device may apply to data processed by the network device, from a second individual having responsibility for managing the network.

14. A method as recited in Claim 1, wherein determining one or more processing policies comprises creating and storing one or more policy statements in a repository, wherein each policy statement associates a condition of one of the traffic flows, an operator, an operand, and an action comprising one of the quality of service treatments.

15. A method as recited in Claim 1, wherein determining one or more processing policies comprises creating and storing one or more policy statements in a repository, wherein each policy statement is represented by a plurality of nodes that represent a condition of one of the traffic flows, an operator, an operand, and an action comprising one of the quality of service treatments.

16. A method as recited in Claim 1, wherein determining one or more processing policies comprises creating and storing one or more policy statements in a directory, wherein each policy statement is represented by a plurality of nodes that represent a condition of one of the traffic flows, an operator, an operand, and an action comprising one of the quality of service treatments, and wherein the plurality of nodes is coupled to a root node having a distinguished name in the directory.

17. A method as recited in Claim 1, wherein each of the mappings comprises an application code point value stored in associated with a differentiated services code point value.

18. (Canceled)

19. A method of selectively establishing a quality of service value treatment for network traffic passing through a particular device in a data network that comprises a plurality of other heterogeneous network devices, according to an application program that generates the network traffic, comprising the steps of:

receiving application information that defines one or more traffic flows associated with one or more message types generated by the application program, including one or more application codepoints at which an application generates the traffic flows;

receiving device information that defines one or more quality of service treatments that the particular network device is capable of applying to data processed by the particular network device;

based on the device information and the application information, determining one or more processing policies that associate the traffic flows with the quality of service treatments;

creating and storing one or more mappings of the application points to the quality of service treatments that may be used with the processing policies to generate the quality of service value when the application program generates traffic flows of one of the message types;

storing the mappings in a repository that is accessible by the application program;

converting the mappings into one or more messages to the network device that instruct the network device to apply Differentiated Services quality of service treatment in response to receiving traffic from the application program that matches the traffic flows;

wherein the step of converting the mappings is performed before transmitting said traffic flows of one of the message types to said network;

enforcing one of the processing policies at the network device in response to receiving traffic from the application program that matches the traffic flow type; and

wherein enforcing one of the processing policies comprises:

requesting, using an application QoS policy element that is coupled to the application program, an operating system function to modify a packet of the traffic flows using a policy element that requests a different operating system function according to the operating system then in use; and

at the network device, in response to receiving traffic from the application program that matches the traffic flow type and in response to the operating system function, modifying a portion of the packet to activate a quality of service treatment of the network device.

20. A method of selectively establishing a quality of service value for a particular network device in a network that comprises a plurality of other heterogeneous network devices, comprising the steps of:

receiving application information that defines one or more traffic flows associated with one or more message types generated by an application program, including information identifying one or more points at which an application generates the traffic flows;

receiving device QoS information that defines one of more quality of service treatments that the particular network device may apply to data processed by the particular network device;

based on the device QoS information and the application information, determining one or more processing policies that associate the traffic flows with the quality of service treatments;

creating and storing one or more mappings of the application points to the quality of service treatments that may be used with the processing policies to generate the quality of service value when the application program generates traffic flows for one of the message types;

causing generation of the quality of service value, wherein the generation of the quality of service value is based on said one or more mappings and is performed before transmitting said traffic flows of one of the message types to said network;

enforcing one of the processing policies at the network device in response to receiving traffic from the application program that matches the traffic flow type; and

wherein enforcing one of the processing policies comprises:

requesting, using an application QoS policy element that is coupled to the application program, an operating system function to modify a packet of the traffic flows using a policy element that requests a different operating system function according to the operating system then in use; and

at the network device, in response to receiving traffic from the application program that matches the traffic flow type and in response to the operating system function, modifying a portion of the packet to activate a quality of service treatment of the network device.

21. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, cause the one or more processors to selectively establish a quality of service value for a particular network device in a network that comprises a plurality of other heterogeneous network devices, by carrying out the steps of:

receiving application information that defines one or more traffic flows associated with one or more message types generated by an application program, including information identifying one or more points at which an application generates the traffic flows;

receiving device information that defines one of more quality of service treatments that the particular network device may apply to data processed by the particular network device;

based on the device information and the application information, determining one or more processing policies that associate the traffic flows with the quality of service treatments;

creating and storing one or more mappings of the application points to the quality of service treatments that may be used with the processing policies to generate the quality of service value when the application program generates traffic flows for one of the message types;

causing generation of the quality of service value, wherein the generation of the quality of service value is based on said one or more mappings and is performed before transmitting said traffic flows of one of the message types to said network;

enforcing one of the processing policies at the network device in response to receiving traffic from the application program that matches the traffic flow type; and

wherein enforcing one of the processing policies comprises:

requesting, using an application QoS policy element that is coupled to the application program, an operating system function to modify a packet of the traffic flows using a policy element that requests a different operating system function according to the operating system then in use; and

at the network device, in response to receiving traffic from the application program that matches the traffic flow type and in response to the operating system function, modifying a portion of the packet to activate a quality of service treatment of the network device.

22. A computer-readable medium as recited in Claim 21, further comprising instructions for carrying out the steps of:

storing the mappings in a repository that is accessible by the application program;

storing both the application information and the device information in the repository; and

converting the mappings into one or more settings of the network device.

23. A computer-readable medium as recited in Claim 21, further comprising instructions for carrying out the steps of:

creating and storing one or more classes that classify the traffic flows, each of the classes associated with one or more of the message types;

based on the device information and the classes of the traffic flows, determining one or more processing policies that associate the traffic flows with the quality of service treatments.

24. A computer-readable medium as recited in Claim 21, further comprising instructions for carrying out the steps of creating and storing one or more mappings by creating and storing one or more policies, concerning network processing of traffic flows generated by the application program, in the repository.

25. A computer-readable medium as recited in Claim 21, further comprising instructions for carrying out the steps of creating and storing one or more mappings by creating and storing one or more policies, concerning network processing of traffic flows generated by the application program, in a policy server coupled to a Lightweight Directory Access Protocol directory that comprises the repository.

26. A computer-readable medium as recited in Claim 21, further comprising instructions for carrying out the steps of creating and storing one or more mappings by creating and storing, in the repository, one or more mappings of Application Code Points of the application program to one or more Differential Services Code Points of a protocol associated with the network device.

27. A computer-readable medium as recited in Claim 21, further comprising instructions for carrying out the steps of determining one or more processing policies by creating and storing one or more policy statements in a repository, wherein each policy statement

associates a condition of one of the traffic flows, an operator, an operand, and an action comprising one of the quality of service treatments.

28. A computer-readable medium as recited in Claim 1, further comprising instructions for determining one or more processing policies by creating and storing one or more policy statements in a directory, wherein each policy statement is represented by a plurality of nodes that represent a condition of one of the traffic flows, and operator, an operand, and an action comprising one of the quality of service treatments, and wherein the plurality of nodes is coupled to a root node having a distinguished name in the directory.

29. A method of selectively establishing a quality of service value for a particular network device in a network that comprises a plurality of other heterogeneous network devices, comprising the steps of:

receiving and storing, in a directory server, application information that defines one or more traffic flows for one or more message types generated by an application program, including information identifying one or more code points at which an application generates the traffic flows;

receiving and storing, in the directory server, device information that defines one of more quality of service treatments that the particular network device may apply to data processed by the particular network device;

based on the device information and the application information, creating and storing a first policy mapping that associates the traffic flows with the quality of service treatments; and

creating and storing a second mapping of the application code points to the quality of service treatments that may be used with the first policy mapping to generate the quality of service value when the application program generates traffic flows for one of the message types;

causing generation of the quality of service value, wherein the generation of the quality of service value is based on said one or more mappings and is performed before transmitting said traffic flows of one of the message types to said network

enforcing one of the processing policies at the network device in response to receiving traffic from the application program that matches the traffic flow type; and

wherein enforcing one of the processing policies comprises:

requesting, using an application QoS policy element that is coupled to the application program, an operating system function to modify a packet of the traffic flows using a policy element that requests a different operating system function according to the operating system then in use; and

at the network device, in response to receiving traffic from the application program that matches the traffic flow type and in response to the operating system function, modifying a portion of the packet to activate a quality of service treatment of the network device.

30. An apparatus for selectively establishing a quality of service value for a particular network device in a network that comprises a plurality of other heterogeneous network devices, comprising:

a network interface that is communicatively coupled to the network for receiving packet flows therefrom;

one or more processors; and

a computer-readable medium carrying one or more sequences of instructions which, when executed by the one or more processors, cause the one or more processors to selectively establish a quality of service value for a particular network device in a network that comprises a plurality of other heterogeneous network devices, by carrying out the methods and steps of any of Claims 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, or 29.